AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application.

1. (Previously Presented) A method for performing distributed simulation, comprising:

providing at least two simulators, wherein at least one of the at least two simulators represents at least one of a component and a system based on processors and chipsets;

providing a backplane having a fixed configuration;

associating an interface with each of the at least two simulators;

interfacing each of the at least two simulators with the fixed configuration backplane via the interface associated with each of the at least two simulators;

exchanging messages between the at least two simulators via the fixed configuration backplane and the associated interfaces; and

operating each interface to convert the messages between a data format associated with the fixed configuration backplane and a data format associated with the simulator associated with the interface.

2. (Original) The method of claim 1, wherein at least one of the at least two simulators is a driver.

3. (Original) The method of claim 1, wherein at least one of the at least two simulators is a checker.

4. (Original) The method of claim 1, wherein at least one of the at least two simulators is a model.

5. (Original) The method of claim 1, wherein the component is an integrated circuit chip.

7. (Original) The method of claim 1, wherein communication between the at least two simulators is based upon one of a central controlling process and a tree.

8. (Cancelled)

9. (Original) The method of claim 1, wherein only the simulators requiring relaxation are relaxed prior to exchanging messages.

10. (Original) The method of claim 1, wherein a system comprising both cycle based simulators and event based simulators is relaxed by holding the cycle based simulators at a designated clock edge until the cycle based simulators are relaxed and by incrementally advancing time along a cycle from the designated clock edge until the event-based simulators are relaxed, the cycle based simulators and the event based simulators being relaxed before advancing a simulation time.

11. (Original) The method of claim 1, further comprising:

executing a deadlock-free remote procedure call function.

12. (Original) The method of claim 11, further comprising:

adding new signals to the backplane global signal vector utilizing the remote procedure call capability.

13. (Previously Presented) The method of claim 1, further comprising:

executing multiple-bit encodings of the exchanged messages to enable simulators based on different encoding schemes to communicate via the simulation backplane.

14. (Original) The method of claim 1, further comprising:

resolving a conflict between a signal strength of messages exchanged via the simulation backplane utilizing a least upper bound operation accomplished by a bitwise-OR operation.

15. (Original) The method of claim 1, wherein messages are exchanged in a manner which accommodate both word-parallel mode capable simulators and non-word-parallel mode capable simulators.

16. (Original) The method of claim 1, further comprising:

executing a protocol such that a simulation backplane least upper bound operation preserves enough information to enable each interface to locally determine and correctly resolve a value of messages exchanged via the simulation backplane.

17. (Original) The method of claim 1, further comprising:

exchanging messages in a manner which accommodate simulators running at different frequencies.

18. (Original) The method of claim 1, wherein design validation tests can be written using High-Level Language (HLL) primitives.

19. (Original) The method of claim 1, wherein one of a same handwritten test and an ATG generated test can be used for all simulators.

20. (Original) The method of claim 1, wherein a protocol checker may be used with an arbitrary collection of simulators.

21. (Previously Presented) A method for performing distributed simulation, comprising:

providing at least two simulators, wherein at least one of the at least two simulators represents at least one of a component and a system based on microprocessors and complex chipsets;

associating an interface with each of the at least two simulators;

interfacing each of the at least two simulators with a simulation backplane via the interface associated with each of the at least two simulators without performing a reconfiguration of the backplane;

exchanging messages between the at least two simulators via the simulation backplane and the associated interfaces;

changing a combination of the at least two simulators and associated interfaces without performing a reconfiguration of the simulation backplane;

wherein each interface converts messages between a data format associated with the backplane and a data format associated with the simulator associated with the interface.

22. (Original) The method of claim 21, wherein at least one of the at least two simulators is a driver.

23. (Original) The method of claim 21, wherein at least one of the at least two simulators is a checker.

24. (Original) The method of claim 21, wherein at least one of the at least two simulators is a model.

25. (Original) The method of claim 21, wherein the component is an integrated circuit chip.

26. (Previously Presented) A method for validating a component, comprising:

sending a test message from a backplane to an interface;

converting the test message from a first data format into a second data format, utilizing the interface;

sending the converted test message from the interface to at least one of the component and at least one model of the component;

receiving, at the interface and in response to the converted test message, a response message from at least one of the component and the at least one model of the component, the response message being in the second data format;

converting the response message from the second data format to the first data format, utilizing the interface;

sending the converted response message to the backplane; and

comparing the converted response message to a predetermined value.

27. (Original) The method of claim 26, wherein the component is an integrated circuit chip.

28. (Original) The method of claim 26, wherein the comparing step is accomplished utilizing a checker.

29. (Previously Presented) A method for validating a component, comprising:

receiving a test message from a first component;

sending the test message to at least one of the component and at least one model of the component via at least one interface which converts the test message from a first data format to a second data format;

receiving a response message from at least one of the component and the at least one model of the component via the at least one interface which converts the response message from the second data format to the first data format;

sending the response message to a second device; and

comparing the response message to a predetermined value utilizing the second device.

30. (Original) The method of claim 29, wherein the component is an integrated circuit chip.

31. (Original) The method of claim 29, wherein the first device is a driver.

32. (Original) The method of claim 29, wherein the first device is one of another component and a model of another component.

33. (Original) The method of claim 29, wherein the second device is a checker.

34. (Previously Presented) A method for validating a component, comprising:

receiving a test message from a backplane;

converting, utilizing an interface, the test message from a first data format to a second data format;

sending the test message to at least one of the component and at least one model of the component;

receiving, in response to the test message, a response message from at least one of the component and the at least one model of the component;

converting, using an interface, the response message from the second data format to the first data format;

sending the response message to a first device via a backplane; and

comparing the response message to a predetermined value utilizing the first device.

35. (Original) The method of claim 34, wherein the component is an integrated circuit chip.

36. (Original) The method of claim 34, wherein the first device is a checker.

37. (Original) The method of claim 34, wherein the test message is received from a second device.

38. (Original) The method of claim 37, wherein the second device is a driver.

39. (Original) The method of claim 37, wherein the second component is one of another integrated circuit chip and a model of another integrated circuit chip.

40. (Original) An apparatus for validating a component, comprising:

a processor;

a computer readable memory segment adapted to be connected to said processor;

an interface module stored within the computer readable memory segment;

a simulation backplane module within the computer readable memory segment, the simulation backplane module comprising computer program code segments which, when executed by the processor, implement the steps of:

receiving a test message from a first device;

sending the test message to at least one of the component and at least one model of the component via the interface module which converts the test message from a first data format to a second data format;

receiving a response message from at least one of the component and the at least one model of the component via the interface module which converts the response message from the second data format to the first data format; and

sending the response message to a second device, the second device comparing the response message to a predetermined value.

41. (Original) The apparatus of claim 40, wherein the component is an integrated circuit chip.

42. (Original) The apparatus of claim 40, wherein the first device is a driver.

43. (Original) The apparatus of claim 40, wherein the first device is one of another component and a model of another component.

44. (Original) The method of claim 40, wherein the second device is a checker.

45. (Previously Presented) An apparatus for validating a component, comprising:

a processor;

a computer readable memory segment adapted to be connected to said processor;

a simulation backplane module within the computer readable memory segment;

an interface module stored within the computer readable memory segment, the interface module comprising computer program code segments which, when executed by said processor, implement the steps of:

receiving a test message from the simulation backplane module;

converting the test message from a first data format to a second data format;

sending the test message to at least one of the component and at least one model of the component;

receiving, in response to the test message, a response message from at least one of the component and the at least one model of the component;

converting the response message from the second data format to the first data format; and

sending the response message to a first device via the simulation backplane module, the first device comparing the response message to a predetermined value.

46. (Original) The apparatus of claim 45, wherein the component is an integrated circuit chip.

47. (Original) The apparatus of claim 45, wherein the first component is a checker.

48. (Original) The method of claim 45, wherein the test message is received from a second device.

49. (Original) The method of claim 48, wherein the second device is a driver.

50. (Original) The method of claim 48, wherein the second device is one of another component and a model of another component.

51. (Previously Presented) A method for performing distributed simulation, comprising:

providing at least two simulators, wherein at least one of the at least two simulators represents at least one of a component and a system based on processors and chipsets;

associating an interface with each of the at least two simulators;

interfacing each of the at least two simulators with a simulation backplane via the interface associated with each of the at least two simulators without performing a reconfiguration of the backplane; and

exchanging messages between the at least two simulators via the simulation backplane and the associated interfaces;

wherein each interface converts messages between a data format associated with the backplane and a data format associated with the simulator associated with the interface.

52. (Previously Presented) An article, comprising:

a storage medium having stored thereon instructions when executed by a processor cause a system to validate a component by performing the following steps:

providing at least two simulators, wherein at least one of the at least two simulators represents at least one of a component and a system based on processors and chipsets;

providing a backplane having a fixed configuration;

associating an interface with each of the at least two simulators;

interfacing each of the at least two simulators with the fixed configuration backplane via the interface associated with each of the at least two simulators;

exchanging messages between the at least two simulators via the fixed configuration backplane and the associated interfaces; and

operating each interface to convert the messages between a data format associated with the fixed configuration backplane and a data format associated with the simulator associated with the interface.

53. (Original) An article, comprising:

a storage medium having stored thereon instructions when executed by a processor cause a system to validate a component by performing the following steps:

receiving a test message from a first component;

sending the test message to at least one of the component and at least one model of the component via at least one interface which converts the test message from a first data format to a second data format;

receiving a response message from at least one of the component and the at least one model of the component via the at least one interface which converts the response message from the second data format to the first data format;

sending the response message to a second device; and

comparing the response message to a predetermined value utilizing the second device.

54. (Previously Presented) An article, comprising:

a storage medium having stored thereon instructions when executed by a processor cause a system to validate a component by performing the following steps:

receiving a test message from a backplane;

converting, utilizing an interface, the test message from a first data format to a second data format;

sending the test message to at least one of the component and at least one model of the component;

receiving, in response to the test message, a response message from at least one of the component and the at least one model of the component;

converting, utilizing an interface, the response message from the second data format to the first data format;

sending the response message to a first device via a backplane; and

comparing the response message to a predetermined value utilizing the first device.

55. (Previously Presented) An article, comprising:

a storage medium having stored thereon instructions when executed by a processor cause a system to validate a component by performing the following steps:

sending a test message from a backplane to an interface;

converting the test message from a first data format into a second data format, utilizing the interface;

sending the converted test message from the interface to at least one of the component and at least one model of the component;

receiving, at the interface and in response to the converted test message, a response message from at least one of the component and the at least one model of the component, the response message being in the second data format;

converting the response message from the second data format to the first data format, utilizing the interface;

sending the converted response message to the backplane; and

comparing the converted response message to a predetermined value.

56. (Previously Presented): The method of claim 1, wherein exchanged messages are gathered together into a global signal vector.